

ROUND DRAWINGS

Contemporary digital practice allows for an incredible facility in the both the generation and fabrication of curves. However, as Michael Young points out, these tools do not bring a “new” geometry to drawing, instead they bring powerful new means of mediation, allowing for “old” geometries to be tested through new configurations, scales, and densities. The circle, for example, is a geometric element whose sheer simplicity renders it seemingly irrelevant in such a plastic environment. A circle is defined by a point rotating about another at a fixed distance, and can be precisely described by most the elementary of mechanical drafting tools, a compass. The simplicity allows for it to be easily dismissed, within the complexities of digital work. However, it is exactly this simplicity that made it such a malleable form in the history of drawing. In this first exercise we will produce a series of iterations of round drawings derived from circular and later spherical coordinates. The intent is to develop a rigorous and imaginative understanding of the generative potentials of drawing with circles.

THE CIRCLE:

Circles will be used throughout the semester. While it is possible to utilize any number of built in software tools, we will begin the semester by developing a simple program that “draws” a circle or an ellipse over time. In so doing, we will take a small step in learning how to construct task specific tools through the use of a programming language. We will also gain more control of a very simple geometric element: the circle. All steps needed to complete this work will be reviewed in class.

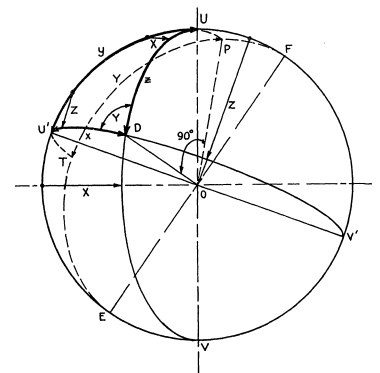
Format(s):

(1) .jpg file @ 1000 x 1000

(1).pyde file

CIRCULAR DRAWINGS:

Creating a circle in most software platforms is as simple as calling the function: Circle (). Inside the parentheses, the user adds values such radius and center (depending on the software). The circle is created as singular graphical object that is static using these variables. It is not as much a circle as it is a representation of a circle. However, we have created a tool that uses the properties of a single point on a circle to create a moving line. We are more concerned with the process of creating the circle than the circle itself. Now we will have to find out what we can do with our new tool. To do so we must first convert the code that we wrote in class into a function. A function encapsulate the several or many lines of code into a single operation. This enables a whole series of other possibilities to emerge. One such possibility is to easily string together several functions in a relatively short program. In our case we will use our new functions to build a drawing that brings together the properties of many circles to produce a new curve that is not circular. In this phase each student will modify their code an produce a set of drawings and animations of these new curves that are not circles.

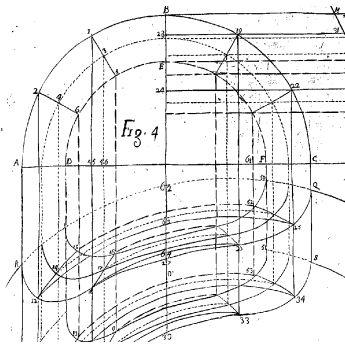


Milton Spheisteing, On Spherical Drawing and Computation, 1955

Modeling software does not bring any “new” geometry, but it does radically change the mediation. (Michael Young, Digital Remediation, 121)

READING:

John May, “Everything Is Already an Image,”
Log, no. 40 (2017): 9–26.



Guarino Guarini, Architettura civile, 1735

Format(s):

- (4) .jpg files @1000 x 1000
- (4) .mp4 files at @1000 x 1000
- (1) .pyde file

THE SPHERE:

The focus of the semester is three-dimensional worlds, and yet so far our world has been flat. We have focused exclusively on the dimensions of x and y. We will now shift to defining objects by three values--x,y, and z--in a three dimensional representation of space. We have actually been using three values the entire time. It is just that our "Z" value has been set to zero. In this next part we will modify our circle function to be a spherical function that defines a point on sphere. We will then use this function to produce a series of drawings of shapes that can be defined by spherical coordinates. In addition, we will begin to look at using color as a means of differentiating aspects of the drawing. Each student must use grey scale line work and (1) additional color.

Format(s):

- (4) .jpg files @1000 x 1000
- (4) .mp4 files at @1000 x 1000
- (1) .pyde file

Spherical Drawings:

This phase is actually a repeat of the Circular Drawing phase. The key difference is that the programs will now use three values. Use the function that we built to describe spherical coordinates to create a set of three-dimensional drawings. These drawings should still be contained within the 1000 pixel x 100 pixel frame, but should explore a range of possible figures defined by spherical coordinates. Each student must use grey scale line work and (1) additional color.

Format(s):

- (4) .jpg files @1000 x 1000
- (4) .mp4 files at @1000 x 1000
- (1) .pyde file